

次世代ものづくりプラットフォーム (HPC/PF)
PDI(パラメータ空間設計・入力支援) サブシステム
操作説明書

Version 1.3.1

2014 年 10 月

改版履歴

リリース	版数	備考
2013/02	1.0	初版
2013/06	1.1	「--no_all」オプションの追加 <cond> ノードの比較演算子の変更
2014/03	1.2	template ファイル記述方式変更 対応ソルバの追加
2014/10	1.3	ワークフローの Lua 化に対応
2014/10	1.3.1	スナップショットファイル名を .pdi_params から snap_params.pdi に変更

目 次

1	インストール	4
1.1	動作環境	4
1.2	必須ソフトウェアのインストール	4
1.2.1	Python のインストール	4
1.2.2	wxPython のインストール	5
1.3	PDI のインストール	6
2	操作方法	8
2.1	プログラムの起動	8
2.2	パラメータ空間設計支援機能	10
2.3	パラメータサーベイ方式設定機能	12
2.4	ソルバ入力用パラメータファイルの生成機能	13
2.4.1	ソルバ入力パラメータテンプレートファイルの指定	13
2.4.2	ソルバ種別の設定	14
2.4.3	サブケースディレクトリとソルバ入力パラメータファイルの生成	14
2.5	パラメータ空間設定情報の保存	16
2.6	バッチモードでの PDI 実行	16
3	ファイルフォーマット	17
3.1	パラメータ定義ファイル	17
3.2	ソルバ入力パラメータテンプレートファイル	25
3.3	スコアファイル	26
3.4	パラメータリストファイル	26
A	外部プログラムの形式	27
A.1	generator プログラム	27
A.2	job suspender プログラム	27

はじめに

本書は、HPC/PF システムの PDI サブシステムについての操作説明書です。

HPC/PF システムは、数値解析シミュレータとその周辺処理ツール群であり、HPC/PF システムを構成するサブシステムの 1 つである PDI は、ユーザのパラメータ空間設計およびパラメータ入力を支援します。

本書では、PDI のインストール方法および操作方法について説明します。

1 インストール

PDI のインストール方法について説明します。

1.1 動作環境

想定される動作環境を以下に示します。

表 2-1 PDI 動作環境

アーキテクチャ	AMD64, IA32
OS	Linux 2.6/3.x MacOS 10.8/10.9 Windows 7/8
ソフトウェア	Python 2.x wxPython 2.7 以降/3.0

1.2 必須ソフトウェアのインストール

1.2.1 Python のインストール

(1) Linux

多くの Linux ディストリビューションでは、Python の実行環境は OS に含まれています。ログイン後にターミナルで「which python」と入力し、Python の実行パスが表示されれば、そのシステムで Python は使用可能です。

Python がインストールされていない場合、各 Linux のディストリビューション管理システムを使用して Python をインストールすることが可能です。以下に例を示します。

RedHat/CentOS 系

```
sudo yum install python
```

Debian/Ubuntu 系

```
sudo apt-get install python
```

(2) MacOS

MacOS 環境向けには、Python のインストーラパッケージが用意されています。

以下に示す URL より、PDI を動作させる環境にあったインストーラをダウンロードし、実行することでインストールをおこなってください。

<http://www.python.org/download/>

尚、同 URL には Python 3.x のインストーラパッケージも登録されていますが、PDI は Python 3.x での動作は保証していません。Python 2.x のインストーラパッケージを使用してください。

(3) Windows

Windows 環境向けには、Python 2.7.x のインストーラパッケージが用意されています。

以下に示す URL より、PDI を動作させる環境にあったインストーラをダウンロードし、実行することでインストールをおこなってください。

<http://www.python.org/download/>

尚、同 URL には Python 3.x のインストーラパッケージも登録されていますが、PDI は Python 3.x での動作は保証していません。Python 2.x のインストーラパッケージを使用してください。

インストーラ実行時に、「Customize Python 2.7.x」という画面が表示され、ここでインストールする Python の構成を選択する事が出来ます。

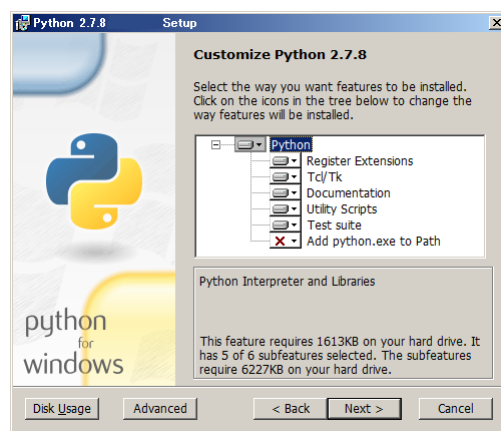


図 1-1 Python Windows インストーラ画面

ここで、デフォルトでは「Add python.exe to Path」という項目が「×」（インストールしない）になっているので、これを「Will be installed on local hard drive」に変更した上でインストールを進めてください。

1.2.2 wxPython のインストール

(1) Linux

多くの Linux ディストリビューションでは、ディストリビューション管理システムを使用して wxPython をインストールすることが可能です。以下に例を示します。

RedHat/CentOS 系

```
sudo yum install wxPython
```

Debian/Ubuntu 系

```
sudo apt-get install python-wxgtk2.8
```

(2) MacOS

MacOS 環境向けには、wxPython のインストーラパッケージが用意されています。以下に示す URL より、PDI を動作させる環境にあったインストーラをダウンロードし、実行することでインストールをおこなってください。

```
http://www.wxpython.org/download.php#stable
```

(3) Windows

Windows 環境向けには、wxPython のインストーラパッケージが用意されています。以下に示す URL より、PDI を動作させる環境にあったインストーラをダウンロードし、実行することでインストールをおこなってください。

```
http://www.wxpython.org/download.php#stable
```

1.3 PDI のインストール

PDI のパッケージは、システムの任意の場所に置くことができます。

PDI のディストリビューションに含まれる `pdi_version.tar.gz` または `pdi_version.zip` ファイル (*version* は実際の文字列に置き換えてください) を任意のディレクトリに展開します。

`tar.gz` の場合は、`tar` コマンドで以下のように実行します。

```
tar xvfz pdi_version.tar.gz
```

展開を実行すると、以下に示すようなディレクトリ階層が作成されます。

```
pdi
  bin
    pdi
    pdi.bat
```

```
doc
    pdi_ug.pdf
    pdi.txt
lib
    python
conf
    PDI.conf
    PDI_log.conf
```

PDIの実行コマンドは、`pdi/bin/pdi`(Linux, MacOS) または `pdi/bin/pdi.bat`(Windows) です。

2 操作方法

PDI の操作方法について説明します。

2.1 プログラムの起動

PDI の実行コマンドファイルは、以下のパスに存在します。

(PDI インストールディレクトリ)/pdi/bin/pdi(.bat)

PDI を起動する際の、コマンドライン指定の形式を以下に示します。。

GUI モード

```
pdi [-x case_directory] [-d param_desc] [-o output_pattern]
    [-t template_file -t template_file ...] [--no_all]
    [-p param_name:param_value -p param_name:param_value ...]
```

バッチモード

```
pdi -b [-x case_directory] [-d param_desc] [-o output_pattern]
    [-t template_file -t template_file ...]
    [-p param_name:param_value -p param_name:param_value ...]
```

引数説明

-b

バッチモード実行。指定された場合、GUI ウィンドウを表示せず、バッチモードで実行します。

-x case_directory

ケースディレクトリ指定。指定されると、PDI は起動時に case_directory にカレントワーキングディレクトリを移動します。指定が省略された場合は、起動ディレクトリとなります。

-d param_desc

パラメータ記述 XML ファイルの指定。パラメータ記述 XML ファイルのパスを指定します。絶対パスか、カレントワーキングディレクトリからの相対パスで指定します。

--no_all

GUI モードにおいて「_All_」ページを作成しません。ただし、パラメータ記述ファイル中に < group > タグが一つも存在しない場合は作成します。

-t template_file

ソルバ入力パラメータテンプレートファイルの指定。入力パラメータテンプレートファイルのパスを指定します。絶対パスか、カレントワーキングディレクトリからの相対パスで指定します。複数指定が可能です。

-o output_pattern

出力先ディレクトリ・ファイル名のパターン指定。サブケース毎に生成する作業ディレクトリ及び作業ディレクトリ配下の入力パラメータファイルのパターンを指定します。パターンは、以下に示す形式で指定します。

directoryname/filename

作業ディレクトリ名と配下のファイル名は「/」で区切られます。パターン中に「/」が現れない場合、すべて filename と解釈され、カレントワーキングディレクトリ直下に filename のパターンで入力パラメータファイルが作成されます。

directoryname が指定された場合、カレントワーキングディレクトリ配下に作業ディレクトリとして directoryname が作成され、その配下に filename が作成されます。

directoryname 中に「%P」が含まれている場合、この部分はスweepされるパラメータ値の組み合わせ文字列に置き換えられます。また、「%Q」が含まれている場合、この部分はスweepされるパラメータ名+パラメータ値の組み合わせ文字列に置き換えられます。

(例)

```
ptest%P/PARAMS      ptest_100_10/PARAMS
ptest%Q/PARAMS      ptest_Re100_CX10/PARAMS
```

パラメータスweepが行われない場合は、「%P」「%Q」は空文字列に置き換えられます。

(例) ptest%P/PARAMS ptest/PARAMS

filename 中に「%T」が含まれている場合、この部分は template_file のベース名に置き換えられます。また、filename 中に「#T」が含まれている場合、この部分は template_file の番号に置き換えられます (template_file が 1 個だけ指定されている場合は「_0」)。

filename 中に「#S」が含まれている場合、この部分は単一ディレクトリ内のサブケース番号 (1 ディレクトリにつき 1 サブケースの場合は、常に 0) に置き換えられます。

directoryname および filename 中に「#D」「#J」が含まれている場合、この部分はそれぞれディレクトリ通番、サブケース通番に置き換えられます。

(例) ptest#D/PARAMS#J ptest_0/PARAMS_0

-p param_name:param_value

パラメータ値の直接指定。パラメータ名とパラメータ値のペアを、コマンドラインから直接指定します。複数指定が可能です。パラメータ名とパラメータ値は「:」で区切って記述します。

パラメータ値は、以下のいずれかの形式で指定を行います。

- 直接指定
- 範囲、刻み幅指定： 最小値/最大値/刻み幅
最小値、最大値、刻み幅を「/」で区切って記述。空白を入れてはならない。
- 列挙： 値 1, 値 2,...
列挙する値を「,」で区切って記述。空白を入れてはならない。

尚、直接指定または列挙で文字列値を指定する場合、空白または区切り文字が含まれる場合は「"」で囲って指定します。

PDI を GUI モードで実行すると、PDI のメインウィンドウが表示されます。

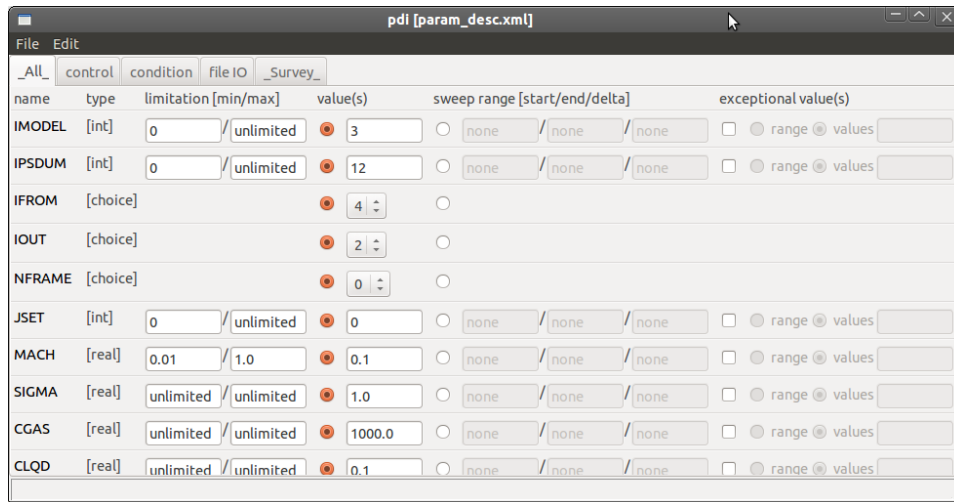


図 3-1 PDI メインウインドウ

尚、PDI の終了は、メインウインドウの File メニューから Quit を選択するか、メインウインドウを閉じることで行えます (MacOS の場合、Quit はメニューバーの「Python」メニューの「Quit Python」です)。

2.2 パラメータ空間設計支援機能

PDI のパラメータ空間設計支援は、パラメータ定義ファイルの記述に基づいて行われます。

パラメータ定義ファイルは、PDI で設定可能なソルバのパラメータの名前、型、値域、デフォルト値、所属グループなどを記述した XML 形式のファイルで、PDI の起動時に「-d」オプションで指定する他、GUI モードの場合は「File」メニューの「load parameter description file」からファイルを指定し、ロードさせることができます。

パラメータ定義ファイルがロードされると、GUI モードの場合パラメータ定義ファイル中に記述された各パラメータ項目は、所属するグループごとにタブページにまとめられ、リスト形式で表示されます。所属グループ属性のないパラメータ項目を含めすべてのパラメータ項目は、「ALL」のタブページにまとめられます。

各パラメータ項目は、パラメータの型に応じてパラメータ値および範囲と刻み幅の入力欄の GUI 部品が配置されます。

name パラメータ名

パラメータ定義ファイル中にパラメータの説明が記述されている場合、マウスカーソルをパラメータ名の上に移動するとパラメータの説明がツールチップとして表示されます。また、パラメータ名をマウス左ボタンでダブルクリックすると、そのパラメータ設定の有効 / 無効が切り替わります。無効の場合はソルバ入力ファイル生成時にそのパラメータの設定は無視され、テンプレートファイルに設定されたデフォルト値が使用されます。パラメータ設定が有効で、かつそのパラメータ空間が縮退している (有効なパラメータ設定値が無い) 場合は、パラメータ名は赤色で表示されます。

type パラメータ型

パラメータ定義ファイル中に記述されたパラメータの型が表示されます。

- int 整数
- real 実数
- choice 候補選択
- string 文字列
- bool 真偽値

limitation 値域

パラメータ値が取りうる値の最小 / 最大値を設定します。制限なしにする場合には、none または unlimited と入力します。type が int または real の場合のみ設定可能です。

value(s) パラメータ値 (単一または列挙)

このラジオボタンが ON の場合には、パラメータ値は直接指定します。type が int または real の場合は、空白で区切って複数の値を列挙指定することができます。

sweep range パラメータスイープ範囲

このラジオボタンが ON の場合には、パラメータ値は範囲と刻み幅で指定します。type が bool の場合は、パラメータ値は True と False の両方でスイープします。type が choice の場合は、全選択肢がパラメータスイープの対象になります。type が string のパラメータは、パラメータスイープの対象にはなりません。

exceptional value(s) 除外値 (範囲または列挙値)

このチェックボックスが ON の場合、パラメータスイープの除外値を設定することができます。range ラジオボタンが ON の場合は、テキストボックスに除外する範囲の最小値と最大値を「/」(スラッシュ) で区切って指定します。values ラジオボタンが ON の場合は、テキストボックスに除外する値を直接指定します。複数の値を除外する場合は、空白で区切って列挙指定することもできます。

尚、除外値の設定を行うことができるのは、type が int または real のパラメータ項目のみです。

設定の例

limitation -0.3 / 0.7 (-0.3 から 0.7 までの値が有効)

sweep range -0.5 / 0.5 / 0.1 (-0.5 から 0.5 まで 0.1 刻みでパラメータスイープ)

exceptional values 0.0 0.1 (0.0 と 0.1 はパラメータスイープから除外)

パラメータスイープされる値: -0.3, -0.2, -0.1, 0.2, 0.3, 0.4, 0.5

(-0.4, -0.5 は limitation から外れるので無効、0.0 と 0.1 は除外)

パラメータ間の依存関係

パラメータ定義ファイル中に、パラメータ間の依存関係 (<depend>タグ) の記述がある場合は、依存先パラメータの値に応じてパラメータ項目欄の状態の変更 (有効/無効、値設定、値域設定) が行われます。

ただし、この状態変更は依存先のパラメータ値が変更された際に行われるので、その後パラメータ欄の操作を行うことでパラメータの状態を再変更することが可能です。

上記の設定において、複数のパラメータ値をとるように設定されたパラメータ項目が 1 個でも存在する場合はパラメータスイープの対象となります。パラメータスイープの総件数は、すべてのパラメータ項目のパラメータ値の件数を掛け合わせた数になります。

2.3 パラメータサーベイ方式設定機能

GUI モードで起動された PDI の「Survey」タブを選択すると、下図に示すパネルが表示され、ここでパラメータサーベイの方式を設定することができます。

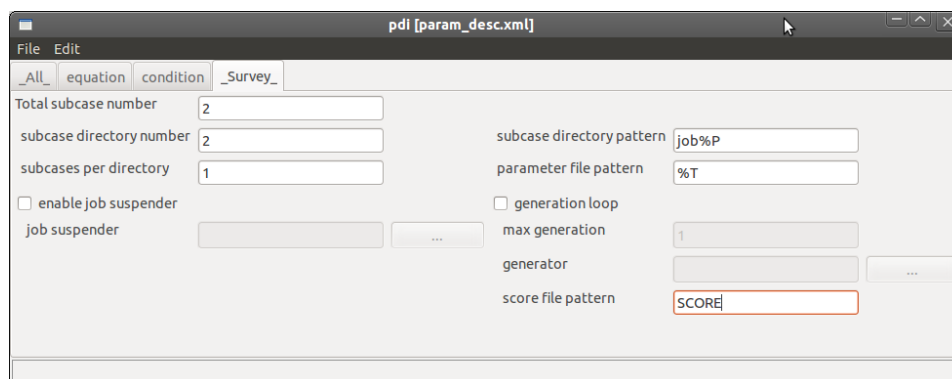


図 3-2 PDI の Survey パネル

全てのパラメータケース (サブケース) を組み合わせた数 (全探索数) が「Total subcase number」に表示されます。この数が 0 の場合は、パラメータ空間が縮退していることを意味し、ソルバ入力用のパラメータファイル生成を行うことはできません。

全探索数に対して、幾つの作業ディレクトリを作成するかを「subcase directory number」に、または 1 つの作業ディレクトリにつき幾つのパラメータケースを割り当てるかを「subcases per directory」に入力します。どちらか一方が入力されると、もう一方は自動的に更新されます。初期値は、1 つの作業ディレクトリにつき 1 つのパラメータケース (作業ディレクトリ数=全探索数) です。 (作業ディレクトリ数) × (作業ディレクトリ当りのパラメータケース数) (全探索数) の場合は、最後の作業ディレクトリに割り当てられるパラメータケース数で調整されます。

作業ディレクトリ名 (subcase directory pattern)、ソルバ入力用パラメータファイル名 (parameter file pattern)、およびスコアファイル名 (score file pattern) については、以下に示すキーワードを使用してパターンを設定します。

%P パラメータ組み合わせ

%Q パラメータ組み合わせ (パラメータ名付き)

%T テンプレートファイルのベース名

(テンプレートファイル名から末尾の ".template", ".tpl", ".tmp", ".tpl" を除いたもの)

#D 作業ディレクトリ通番

#J サブケース通番

#T ソルバ入力パラメータファイルが複数ある場合のファイル番号

尚、全探索数が 1 の場合は、作業ディレクトリ名中の上記パターンは全て空文字列に置き換えられ、末尾に「_0」が追加されます。

ジョブの刈り込みを行う場合、「enable job suspender」のチェックを ON にします。この場合、「job suspender」にジョブの刈り込みを行うユーザプログラムのパスを設定します。

繰り返し型のパラメータサーベイを行う場合、「generation loop」のチェックを ON にします。この場合、「max generation」に最大世代 (ループ) 数を、「generator」に繰り返しチェックと次世代のパラメータ空間を生成するユーザプログラムのパスを設定します。

尚、ジョブの刈り込みまたは繰り返し型のパラメータサーベイを行う場合、ソルバが生成するスコアファイル名のパターンが「score file pattern」に設定されていなければなりません。空白の場合はデフォルトで「SCORE」というファイル名パターンが指定されたとみなされます。

2.4 ソルバ入力用パラメータファイルの生成機能

PDI によるソルバ入力用パラメータファイルの生成を行うには、パラメータ空間設定 (3.2 章) とパラメータサーベイ方式設定 (3.3 章) に加えて、ソルバ入力パラメータテンプレートファイルの指定と、ソルバ種別の指定が必要です。

2.4.1 ソルバ入力パラメータテンプレートファイルの指定

ソルバ入力パラメータテンプレートファイルは、PDI の起動時に「-t」オプションで指定することができます。「-t template.file」という形式で、複数指定することが可能です。

PDI を GUI モードで起動した場合、「Edit」メニューの「set parameter template file(s)」から、ソルバ入力用パラメータテンプレートファイルの追加 / 削除を行うことができます。

このメニューを選択すると、下図に示すダイアログウィンドウが表示されます。

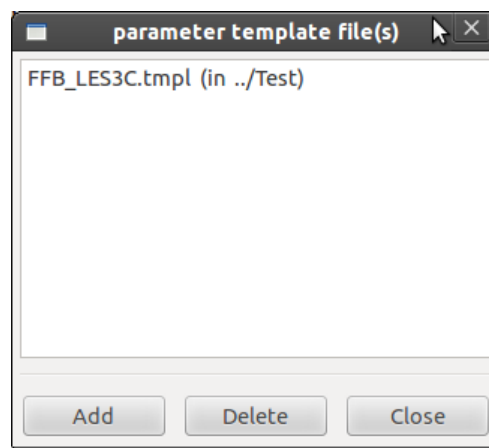


図 3-3 PDI のテンプレートファイル設定ダイアログ

このダイアログウィンドウには、既に登録済みのソルバ入力用パラメータテンプレートファイルがリスト表示されています。このダイアログは「Close」ボタンをクリックすると閉じられます。

「Add」ボタンをクリックすると、ファイル選択ダイアログが表示され、ここで選択したファイルがソルバ入力用パラメータテンプレートファイルとして追加されます。

リスト上で登録済みのソルバ入力用パラメータテンプレートファイルを選択状態にし、「Delete」ボタンをクリックすると、そのファイルの登録は解除されます。

2.4.2 ソルバ種別の設定

PDI では、ソルバ入力用パラメータファイルの生成時に、パラメータスイープ実行のための設定ファイルを作成します。この設定ファイル中には、ソルバ実行の方法の記述も含まれるため、PDI 上でソルバ種別の設定を行なっておく必要があります。

GUI モードで起動した PDI の「Edit」メニューから「select solver type」を選択すると、下図に示すダイアログウィンドウが表示されます。

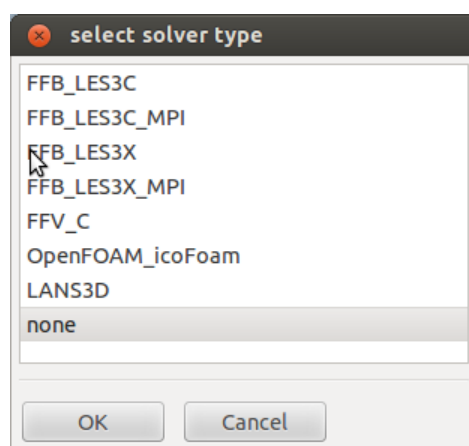


図 3-4 PDI のソルバ種別選択ダイアログ

ここには、PDI に予め登録されているソルバ種別の一覧がリスト表示され、リスト中の項目を選択状態にして「OK」ボタンをクリックすると、ソルバ種別の設定がおこなわれます。

現在、PDI に登録されているソルバ種別の設定を以下に示します。

ソルバ種別	ソルバ実行コマンド	ソルバ入力パラメータファイル名パターン
FFB_LES3C	run_les3c.sh	PARMLES3C
FFB_LES3C_MPI	run_les3c_mpi.sh	PARMLES3C
FFB_LES3X	run_les3x.sh	PARMLES3X
FFB_LES3X_MPI	run_les3x_mpi.sh	PARMLES3X
FFV	run_ffv.sh %T	%T
OpenFoam_icoFoam	run_openfoam.sh	
その他 (none)	run.sh	変更せず

上記の表中の「%T」は テンプレートファイルのベース名 (3.3 章参照) を表します。

また、パラメータスイープ実行のための設定ファイルにおいては、上記の表の「ソルバ実行コマンド」に示された実行ファイル (シェルスクリプト) がケースディレクトリ配下に存在することを前提としています。

2.4.3 サブケースディレクトリとソルバ入力パラメータファイルの生成

GUI モードで起動された PDI では、「File」メニューの「generate solver parameter file(s)」を選択すると、パラメータスイープ用の作業ディレクトリ (サブケースディレクトリ) と、ソルバ入力用パラメータファイルの生成が行われます。

具体的には、以下に示す処理が行われます。

(1) サブケースディレクトリの作成

ケースディレクトリ配下に、作業ディレクトリ名パターンを展開したディレクトリが「subcase directory number」個作成されます。作成しようとしているディレクトリが既に存在する場合は、そのディレクトリはそのまま流用されます。

(2) ソルバ入力用パラメータファイルの作成

作成したサブケースディレクトリの配下に、ソルバ入力用パラメータファイル名パターンを展開したファイル名で、ソルバ入力用パラメータファイルが作成されます。PDI は、指定されているソルバ入力パラメータテンプレートファイル中のマクロ記述をパラメータ値に置換することで、ソルバ入力用パラメータファイルを生成します。複数のンプレートファイルが指定されている場合、それらすべてに対応するソルバ入力用パラメータファイルが作成されます。

作成しようとしているソルバ入力用パラメータファイルが既に存在する場合は、そのファイルは上書きされます。

(3) ワークフロー実行用パラメータ設定ファイルの作成

実際のソルバ実行を含むパラメータスイープ / パラメータサーベイの実行は、ケースワークフローから Lua スクリプトとして実行されます。

PDI は、ソルバ入力用パラメータファイルの生成時に、Lua スクリプトファイルとして、ケースディレクトリ配下に paramsweep_wf.lua というファイルを作成します。同名のファイルが既に存在する場合、そのファイルは上書きされます。

(4) パラメータリストファイルの作成

パラメータリストファイルは、現在のパラメータ空間設定の下で実行されるパラメータスイープのパラメータ値およびスコアファイルのパスを記述した CSV 形式のテキストファイルです。

PDI は、ソルバ入力用パラメータファイルの生成時に、パラメータリストファイルをケースディレクトリ配下に param_list.csv というファイル名で作成します。同名のファイルが既に存在する場合、そのファイルは上書きされます。

(5) generator、job suspender プログラムパスのエクスポート

「enable job suspender」および「generation loop」が ON の場合、「job suspender」および「generator」に設定されたプログラムの実行パスを、以下に示すファイルに出力します。

```
job suspender :
    ケースディレクトリ/job_suspender_prog
generator :
    ケースディレクトリ/generator_prog
```

尚、「enable job suspender」および「generation loop」が OFF の場合は、上記ファイルが存在していれば、ファイルを削除します。

2.5 パラメータ空間設定情報の保存

PDI は終了時に、その時点でのパラメータ空間設定情報とパラメータサーベイ方式設定情報を、ケースディレクトリ配下に `snap_params.pdi` というファイル名で保存します。既にこのファイルが存在する場合、ファイルは上書きされます。

`snap_params.pdi` ファイルが存在するディレクトリがケースディレクトリとして指定 (`-x`) された場合、パラメータ記述ファイルの指定 (`-d`) がなくても PDI はパラメータ空間設計を行うことが可能です (この場合、ログにはワーニングが出力されます)。また、パラメータ記述ファイルの指定が行われた場合は、パラメータ記述ファイルのロード後に `snap_params.pdi` ファイルに保存された情報の反映を行います。

ただし、指定されたパラメータ記述ファイルと、`snap_params.pdi` ファイルに保存されているパラメータ記述ファイル名が異なる場合は、`snap_params.pdi` ファイルの内容は無視されます。

2.6 バッチモードでの PDI 実行

PDI 起動時に「`-b`」オプションを指定すると、PDI は GUI ウィンドウを表示せず、バッチモードで実行されます。

バッチモードでの PDI 実行は主に、以前に設定されたパラメータスイープにおけるパラメータ空間の変更を行うために行います。PDI 起動時に「`-p`」オプションで 1 個ないし複数個のパラメータ値を変更することにより、ケースディレクトリ/`snap_params.pdi` ファイルに保存された前回設定のパラメータ空間が更新されます。ただしこの場合は、前回パラメータ空間を設定した際と同じケースディレクトリが指定されていることが必要です。

実行の例

```
pdi -x case01 -p Re:100/200/10
```

(ケースディレクトリ「`case01`」において、「`Re`」という名前のパラメータを、100 から 200 まで 10 刻みでスイープさせる)

PDI の「`-p`」オプション指定方法の詳細は、3.1 章を参照してください。

3 ファイルフォーマット

3.1 パラメータ定義ファイル

パラメータ定義ファイルは、XML 形式のテキストファイルとして記述されます。

以下に、構成要素となる XML ノードの記述方式を示します。

(1) トップレベルノード

パラメータ定義ファイルのトップレベルノードは<hpcpf_paramdesc>です。

このノード配下に記述された内容だけが PDI の処理対象となります。

[記述形式]

```
<hpcpf_paramdesc [classification="snapshot"]>
...
</hpcpf_paramdesc>
```

<hpcpf_paramdesc>ノードには、classification 属性としてキーワード"snapshot"を記述することができます。このキーワードが記述されたパラメータ定義ファイルは、パラメータ空間設定情報を保存したスナップショットデータファイルと認識されます。スナップショットデータファイルは通常、PDI の終了時にケースディレクトリ配下に snap_params.pdi というファイル名で作成されます。

(2) パラメータ記述ノード

パラメータ項目の記述には、<param> ノードを使用します。

[記述形式]

```
<param name="パラメータ名" type="パラメータタイプ" [disable="True|False"]>
    [パラメータ説明]
    [サブノード群]
</param>
```

<param>ノードには、name 属性および type 属性を記述する必要があります。

name 属性には、パラメータ名を記述します。

type 属性には、以下のいずれかのパラメータタイプを記述します。

```
int 整数
real 実数
bool 真偽値 (True | False)
string 文字列
choice 文字列候補
```

<param> ノードには、disable 属性を記述することができます。disable 属性には"True"または"False"のキーワードを記述し、True の場合はそのパラメータは無効化されます。記述を省略した場合、disable 属性は"False"とみなされます。

<param> ノード配下には、パラメータ説明のテキストと、複数のサブノードを記述することができます。

<param> ノード配下の最初のテキストは、パラメータ説明の記述として扱われます。

<param> ノード配下に記述できるサブノードについては、以下の説明を参照して下さい。

(3) パラメータ記述サブノード

以下は、<param> ノード配下に記述できるサブノードの説明です。

(3-a) <item> ノード

<item> ノードは、<param> ノードの type 属性が"choice"の場合の、文字列候補の 1 つを記述するノードで、<param> ノード配下に複数 (選択候補数) 個記述できます。

[記述形式]

```
<item>文字列候補 1</item>
```

```
<item>文字列候補 2</item>
```

<param> ノードの type 属性が"choice"意外の場合は、<item> ノードの記述は無視されます。

(3-b) <range> ノード・<minmax> ノード

<range> ノードおよび<minmax> ノードは、<param> ノードの type 属性が"int"または"real"の場合の、値域を記述するノードです。

[記述形式]

```
<range min="最小値" max="最大値" />
```

```
<minmax min="最小値" max="最大値" />
```

min 属性に値域の最小値を、max 属性に最大値を記述します。min 属性またはmax 属性の記述を省略した場合、省略された方向の値制限は無しになります。また、<range> ノードおよび<minmax> ノードの記述自体を省略すると、値域制限は無しになります。

(3-c) <value> ノード

<value> ノードは、パラメータ値の初期値を記述するノードです。

[記述形式]

```
<value>パラメータ値</value>
```

<value> ノードの記述を省略した場合、パラメータ値の初期値は<param> ノードの type 属性に応

じ、以下に示すようになります。

```
int "0"

real "0.0"

bool "False"

string "" (空文字列)

choice "0"(先頭の候補)
```

<param>ノードの type 属性が"int"または"real"の場合は、パラメータ値は空白で区切って複数の値を列挙することができます。

(3-d) <useRange>ノード

<useRange>ノードは、<param>ノードのパラメータ値を値域・刻み幅指定で定義するかどうかを記述するノードです。

[記述形式]

```
<useRange> True | False </useRange>
```

"False"の場合、<value>ノード記述に従ってパラメータ値を設定します。

"True"の場合、<value>ノードの記述は参照せず、次項で説明する<sweepRange>ノードの記述に従ってパラメータ値を設定します。ただし、<param>ノードの type 属性が"bool"の場合はパラメータ値が"True"と"False"の2サブケースとして、<param>ノードの type 属性が"choice"の場合は<item>ノードで記述した全ての候補をパラメータ値とするサブケースが設定されます。<param>ノードの type 属性が"string"の場合、<useRange>ノードの値は無視され、常に<value>ノードの記述に従ってパラメータ値が設定されます。

<useRange>ノードの記述を省略した場合のデフォルト値は"False"です。

(3-e) <sweepRange>ノード

<sweepRange>ノードは、<useRange>ノードの値が"True"の場合の、値域・刻み幅指定によるパラメータ値の設定を記述するノードです。

[記述形式]

```
<sweepRange min="最小値" max="最大値" delta="刻み幅" />
```

min 属性に値域の最小値を、max 属性に最大値を、delta 属性に刻み幅を記述します。これらの属性の記述は省略できません。

尚、<sweepRange>ノードの記述は<param>ノードの type 属性が"int"または"real"の場合以外は無視されます。

(3-f) <useExcept>ノード

<useExcept>ノードは、<param>ノードのパラメータ値に対する除外値指定を参照するかどうかを記述するノードです。

[記述形式]

```
<useExcept> True | False </useExcept>
```

"False"の場合、除外値の指定は参照されません。

"True"の場合、次項以降で説明する<except>ノードまたは<exceptRange>ノードの記述に従って除外値を設定します(最後に記述された<except>ノードまたは<exceptRange>ノードが有効になります)。

(3-g) <except>ノード

<except>ノードは、<param>ノードのパラメータ値に対する除外値を直接記述するノードです。

[記述形式]

```
<except>除外値</except>
```

除外値は、空白で区切って複数指定することが可能です。尚、<except>ノードの記述は<param>ノードのtype属性が"int"または"real"の場合以外は無視されます。

(3-h) <exceptRange>ノード

<exceptRange>ノードは、<param>ノードのパラメータ値に対する除外値を範囲で指定するノードです。

[記述形式]

```
<except min="最小値" max="最大値" />
```

min属性に除外範囲の最小値を、max属性に最大値を記述します。min属性またはmax属性の記述を省略した場合、省略された方向の除外範囲制限は無しになります。

尚、<exceptRange>ノードの記述は<param>ノードのtype属性が"int"または"real"の場合以外は無視されます。

(3-i) 依存関係記述ノード

<param>ノード配下には、他のパラメータ項目との依存関係を示すためのサブノードを記述することができます。依存関係記述ノードの記述方法については、(5)を参照して下さい。

(4) パラメータのグルーピング

複数のパラメータ項目をグループとしてまとめるため、<group>ノードを記述することができます。

[記述形式]

```
<group name="グループ名">
```

[パラメータノード群]

```
</group>
```

<group> ノードには、name 属性を記述する必要があります。name 属性には、グループ名を記述します。

<group> ノード配下には、グルーピングするパラメータ項目の<param> ノードを複数記述することができます。グルーピングされたパラメータ群は、PDI サブシステム上でパラメータ値の設定を行う際に、まとめて参照出来るように処理されます。

(5) 依存関係の記述

(5-a) <depend> ノード

<depend> ノードは、この ノードを配下に持つ<param> ノード（親 ノード）の記述が、他の<param> ノードのパラメータ値に依存することを示すためのノードです。

[記述形式]

```
<depend target="パラメータ項目名" [target2="パラメータ項目名 2"]>
  <cond> ノード
</depend>
```

<depend> ノードには、依存する<param> ノードの名前 (name 属性) を target および target2 属性に記述する必要があります。ここに記述した<param> ノードが存在しない場合、この<depend> ノードの記述は無視されます。target 属性は必ず記述する必要がありますが、target2 属性の記述は無くても構いません。

<depend> ノード配下には、依存する<param> ノードとの関係条件を記述する<cond> ノードをサブノードとして記述します。

尚、<depend> ノードは<param> ノードの配下にのみサブノードとして記述することができます。

(5-b) <cond> ノード

<cond> ノードは、パラメータ項目間の依存関係の条件を記述するためのノードで、<depend> ノードのサブノードとして記述します。

[記述形式]

```
<cond> 「条件式」 ? 「真文」 : 「偽文」 </cond>
```

ここで、「条件式」は以下の形式で記述します。

```
「ターゲット」「演算子」「値」
「ターゲット」「演算子」「値」「結合演算子」「ターゲット 2」「演算子 2」「値 2」
```

「ターゲット」には固定文字列として"VAL"を記述します。

「演算子」には、以下のいずれかを記述します。

「==」または「EQ」

「!=」または「NE」

「<」または「LT」

「<=」または「LE」

「>」または「GT」

「>=」または「GE」

(一部の Python/XML 処理系では、XML タグ以外で「<」文字を使用出来ません。)

<cond> ノードの上位の <depend> ノードで target2 属性が指定されている場合、ここで指定された <param> ノードの値を「値 2」として、2 つの条件式を「結合演算子」で結合する事が出来ます。

「結合演算子」以下のいずれかを記述します。

「&&」または「AND」

「||」または「OR」

ターゲットに指定されたパラメータの type 属性が "string" または "choice" の場合、記述できる演算子は「==」、「!=」、「<」および「>」のみで、「==」が文字列の一致を、それ以外は文字列の不一致を表します。

「真文」および「偽文」には、以下のいずれかを記述することができます。

「enable」

「disable」

「set value="値"」

「set range="最小値 最大値"」

「set sweep="最小値 最大値 刻み幅"」ここで、「enable」および「disable」は、このパラメータ項目の有効化 / 無効化を表します。また、「set value」、「set range」および「set sweep」は、このパラメータ項目の値、値域およびパラメータスイープ範囲と刻み幅を設定 (変更) することを意味します。

(6) パラメータサーベイ方式設定情報

パラメータサーベイ方式の設定情報は、<survey> ノードに記述されます。

[記述形式]

```
<survey>
```

```
  [サブノード群]
```

```
</survey>
```

<survey> ノード配下に記述できるサブノードについては、以下の説明を参照して下さい。

(6-a) <wdPattern> ノード

<wdPattern> ノードは、サブケースジョブの作業ディレクトリ名のパターンを記述するノードです。

[記述形式]

```
<wdPattern>作業ディレクトリ名パターン</wdPattern>
```

(6-b) <pfPattern> ノード

<pfPattern> ノードは、サブケースジョブのソルバ入力パラメータファイル名のパターンを記述するノードです。

[記述形式]

```
<pfPattern>ソルバ入力パラメータファイル名パターン</pfPattern>
```

(6-c) <sfPattern> ノード

<sfPattern> ノードは、サブケースジョブのスコアファイル名のパターンを記述するノードです。

[記述形式]

```
<sfPattern>スコアファイル名パターン</sfPattern>
```

(6-d) <enableSuspend> ノード

<enableSuspend> ノードは、パラメータサーベイにおけるジョブの刈り込みを有効にするかどうかを記述するノードです。

[記述形式]

```
<enableSuspend> True | False </enableSuspend>
```

"True"が記述された場合、ジョブの刈り込みを有効に設定します。

(6-e) <jobSuspend> ノード

<jobSuspend> ノードは、パラメータサーベイにおけるジョブの刈り込みを行う job suspender プログラムの実行パスを記述するノードです。

[記述形式]

```
<jobSuspend>job suspender プログラムの実行パス</jobSuspend>
```

job suspender プログラムの実行パスは、絶対パスか、ケースディレクトリからの相対パスで指定します。

(6-f) <generationLoop> ノード

<generationLoop> ノードは、パラメータサーベイにおける世代ループを有効にするかどうかを記述するノードです。

[記述形式]

```
<generationLoop> True | False </generationLoop>
```


"True"が記述された場合、世代ループを有効に設定します。

(6-g) <generator> ノード

<generator> ノードは、パラメータサーベイにおける世代ループで、各サブケースの評価値を解釈し、次の世代のパラメータスイープ用のパラメータセットを生成する generator プログラムの実行パスを記述するノードです。

[記述形式]

```
<generator>generator プログラムの実行パス</generator>
```

generator プログラムの実行パスは、絶対パスか、ケースディレクトリからの相対パスで指定します。

(6-h) <maxGeneration> ノード

<maxGeneration> ノードは、パラメータサーベイにおける世代ループの最大世代数を記述するノードです。

[記述形式]

```
<maxGeneration>最大世代数</maxGeneration>
```

最大世代数に 0 以下の値が設定された場合、最大世代数の制限は無しとみなされます。

<maxGeneration> ノードの記述を省略した場合のデフォルト値は 1 です。

(6-i) <curGeneration> ノード

<curGeneration> ノードは、パラメータサーベイにおける世代ループの、現在の世代番号を記述するノードです。

[記述形式]

```
<curGeneration>世代番号</curGeneration>
```

初代の世代番号は 0 で、負値が指定された場合は無効となります。

<curGeneration> ノードの記述を省略した場合のデフォルト値は 0 です。

(7) PDI 設定情報

上記以外の PDI の設定情報は、<snapshot> ノードに記述されます。

[記述形式]

```
<snapshot>
```

[サブノード群]

```
</snapshot>
```

<snapshot> ノード配下に記述できるサブノードについては、以下の説明を参照して下さい。

(7-a) <desc_path> ノード

<desc_path> ノードは、PDI にロードされたパラメータ定義ファイルのパスを記述するノードです。

[記述形式]

<desc_path>パラメータ定義ファイルのパス</desc_path>

パラメータ定義ファイルのパスは絶対パスか、ケースディレクトリからの相対パスで指定します。尚、<snapshot> ノード配下に複数の<desc_path> ノードが記述された場合、最後に記述された<desc_path> ノードが有効になります。

(7-b) <templ_path> ノード

<templ_path> ノードは、PDI に登録されたテンプレートファイルのパスを記述するノードです。

[記述形式]

<templ_path>テンプレートファイルのパス</templ_path>

テンプレートファイルのパスは絶対パスか、ケースディレクトリからの相対パスで指定します。<templ_path> ノードは<snapshot> ノード配下に複数記述することができます。

3.2 ソルバ入力パラメータテンプレートファイル

ソルバ入力パラメータテンプレートファイルは、ソルバ入力パラメータファイルの雛形であり、PDI で設定されたパラメータ値を埋め込むためのマクロ記述を含むものです。

以下に、テンプレートファイル中のマクロ記述形式を示します。

%パラメータ名 [! デフォルト値] [* 数値 1] [+ 数値 2] %

PDI による入力パラメータファイル生成時に、「%パラメータ名%」は PDI 上で設定されたパラメータ値に置き換えられます。

ここで、「パラメータ名」はパラメータ定義ファイルの<param> ノードの名前 (name 属性値) で、デフォルト値はパラメータ定義ファイル中に name 属性が「パラメータ名」である<param> ノードが存在しない場合のパラメータ値です。デフォルト値は (「!」を含めて) 記述を省略することができます。

パラメータ項目のタイプが "int" または "real" の場合には、マクロ中に「演算子」と数値を記述することができます。これは、PDI 上で設定されたパラメータ値に対して一次変換を適用した結果を入力パラメータファイルに記述するためのもので、省略が可能です。入力パラメータファイルに記述される最終的な値は、以下のようになります。

(PDI で設定されたパラメータ値) × (数値 1) + (数値 2)

パラメータ項目のタイプが"bool"の場合にも、マクロ中に「演算子」と数値を記述することができます。この場合、「数値 1」部分はパラメータ値が True の場合の記述文字列、「数値 2」部分はパラメータ値が False の場合の記述文字列と解釈され、省略時にはそれぞれ"True", "False"になります。

3.3 スコアファイル

スコアファイルは、パラメータサーベイにおける各サブケースの評価値を格納するテキストファイルで、各サブケースディレクトリに以下に示す形式で作成されます。

```
# コメント
123.456
```

行頭が「#」である行はコメントとして無視します。

評価値は単一の実数値であり、コメント行を除く最初の行に記述された最初の実数値が評価値として扱われます。評価値はパラメータサーベイにおける、ある指標を表すものであり、この数値の優劣の解釈は generator プログラムおよび job suspender プログラムによって異なります。HPC/PF システムとしては、評価値について優劣の規定 (大きい値が優れている、等) は行いません。

3.4 パラメータリストファイル

パラメータリストファイルは、現在実行されているパラメータスイープのパラメータ値およびスコアファイルのパスを記述したファイルです。以下に示す情報が格納された CSV 形式のテキストファイルで、デフォルトでは「ケースディレクトリ/param_list.csv」に置かれます。

```
1 行目 現在の世代番号, 最大世代数
2 行目 パラメータ項目数, パラメータ名 1, パラメータ名 2, ..., パラメータ名 N
3 行目 スコアファイル名, パラメータ値 1, パラメータ値 2, ..., パラメータ値 N
...
```

パラメータリストファイルの 1 行目には、現在の世代番号 (第 1 世代は 0) と最大世代数が格納されます。最大世代数が記述されていない場合は、世代数の制限なしを意味します。

2 行目には、パラメータ項目数 (=N) と、スイープするパラメータ名 (N 個) が記述されます。

3 行目以降には、各サブケース毎のスコアファイル名 (サブケースディレクトリ名/スコアファイル名) と、そのサブケースにおけるパラメータ値 (N 個) が記述されます。

A 外部プログラムの形式

A.1 generator プログラム

generator プログラムは、パラメータサーベイにおける各サブケースの評価値を解釈し、次の世代のパラメータスイープ用のパラメータセットを生成するプログラムです。

HPC/PF システムでは generator プログラムのインターフェースのみを定義します。使用者は下記の仕様に則って generator プログラムを実装する必要があります。

コマンドライン形式

generator プログラムファイル [-x case_directory] [-l param_list.csv]

引数説明

-x case_directory ケースディレクトリ指定

指定されると、generator プログラムは起動時に、case_directory にカレントワーキングディレクトリを移動します。指定が省略された場合は、起動されたディレクトリとなります。

-l param_list.csv パラメータリストファイルの指定

パラメータリストファイルのパスを指定します。絶対パスか、ケースディレクトリからの相対パスで指定します。省略時は、ケースディレクトリ/param_list.csv を指定したとみなされます。

戻り値 (終了ステータス)

- 1 正常終了 (次世代のパラメータスイープを実行する)
- 0 正常終了 (次世代のパラメータスイープを実行しない)
- 1 異常終了

A.2 job suspender プログラム

job suspender プログラムは、単一世代のパラメータスイープにおいて、既に実行が終了したサブケースジョブの結果から、実行する必要のないサブケースジョブの抽出と、そのサブケースジョブのキャンセルを行うプログラムです。

HPC/PF システムでは job suspender プログラムのインターフェースのみを定義します。使用者は、下記の仕様に則って job suspender プログラムを実装する必要があります。

コマンドライン形式

job suspender プログラムファイル [-x case_directory] [-l param_list.csv]

引数説明

-x case_directory ケースディレクトリ指定

指定されると、generator プログラムは起動時に、case_directory にカレントワーキングディレクトリを移動します。指定が省略された場合は、起動されたディレクトリとなります。

-l param_list.csv パラメータリストファイルの指定

パラメータリストファイルのパスを指定します。絶対パスか、ケースディレクトリからの相対パスで指定します。省略時は、ケースディレクトリ/param_list.csv を指定したとみなされます。

戻り値 (終了ステータス)

0 正常終了

0 以外 異常終了

job suspender プログラムが参照するパラメータリストファイルは、generator プログラムが参照するものと同じです (前章を参照)。ただし、generator プログラムの場合と異なり、job suspender プログラムの実行時は現在の世代のパラメータスイープが実行中であり、パラメータリストファイルに記述されているスコアファイルが存在しないサブケースジョブがある点に留意が必要です。